

Violation Learning Differential Evolution-Based hp-Adaptive Pseudospectral Method for Trajectory Optimization of Space Maneuver Vehicle

RUNQI CHAI, Student Member, IEEE
AL SAVVARIS, Member, IEEE
ANTONIOS TSOURDOS, Member, IEEE
Cranfield University, Cranfield, U.K.

The sensitivity of the initial guess in terms of optimizer based on an hp-adaptive pseudospectral method for solving a space maneuver vehicle's (SMV) trajectory optimization problem has long been recognized as a difficult problem. Because of the sensitivity with regard to the initial guess, it may cost the solver a large amount of time to do the Newton iteration and get the optimal solution or even the local optimal solution. In this paper, to provide the optimizer a better initial guess and solve the SMV trajectory optimization problem, an initial guess generator using a violation learning differential evolution algorithm is introduced. A new constraint-handling strategy without using penalty function is presented to modify the fitness values so that the performance of each candidate can be generalized. In addition, a learning strategy is designed to add diversity for the population in order to improve the convergency speed and avoid local optima. Several simulation results are conducted by using the combination algorithm; simulation results indicated that using limited computational efforts, the method proposed to generate initial guess can have better performance in terms of convergence ability and convergence speed compared with other approaches. By using the initial guess, the combinational method can also enhance the quality of the solution and reduce the number of Newton iteration and computational time. Therefore, the method is potentially feasible for solving the SMV trajectory optimization problem.

Manuscript received December 31, 2015; revised August 1, 2016 and December 11, 2016; released for publication February 5, 2017. Date of publication March 9, 2017; date of current version August 7, 2017.

DOI. No. 10.1109/TAES.2017.2680698

Refereeing of this contribution was handled by I. Hwang.

Authors' address: R. Chai, A. Savvaris, and A. Tsourdos are with the School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield MK43 0AL, U.K. E-mail: (r.chai@cranfield.ac.uk; a.savvaris@cranfield.ac.uk; a.tsourdos@cranfield.ac.uk).

0018-9251/16 © 2017 CCBY

I. INTRODUCTION

Trajectory optimization problems for re-entry vehicles have been investigated widely by some researchers [1]–[4]. One of the current objectives is the development of space maneuver vehicles (SMV) for a dynamic mission profile. Due to the high nonlinear nature of the problem, numerical methods are usually employed to solve this type of problems [5], [6]. Numerical methods for solving optimal control problems are divided into two major classes: indirect methods and direct methods [7], [8]. However, it is more challenging to solve the re-entry problem by using indirect methods, since in an indirect method, first-order necessary conditions for optimality should be derived from the original optimal control problem according to the calculus of variations so that the Hamiltonian boundary-value problem can be constructed. This process usually becomes costly due to the complexity of the dynamic model and path constraints. Hence, the direct optimization method has been widely used in addressing trajectory optimization type of problems.

It is worth noting that other algorithms were also proposed such as that by Liu *et al.* [9], [10] who proposed specific convex programming methods. These methods are based on second-order cone programming for solving general trajectory optimization problems. By applying these algorithms, the original problem is first convexified into a convex optimal control problem, and then discretization is applied to generate a second-order cone programming problem.

In the investigation carried out in this paper, the direct method is applied to solve the problem, in which the first step is to parameterize the continuous-time optimal control problem. For the discretization techniques, the most common methods are multiple shooting, direct collocation, and pseudospectral methods [7], [11]–[17]. Using these methods, the optimal control problem can be transformed to a nonlinear optimization problem or nonlinear programming problems (NLPs) [18]. The resulting NLP problem can be solved numerically by well-developed algorithms such as sequential quadratic programming (SQP) and interior point method. Most of the discretization techniques can provide very good accuracy for many cases. Take the multiple shooting method as an example, if the dynamics are parameterized with fix-step, the high accuracy can be achieved if the step length is small enough [19]. In recent years, pseudospectral methods have attracted more attention and more work is being carried out in this field [6]. The main advantage with pseudospectral methods [20], [21] is that high accuracy can be achieved with much less temporal nodes [22], [23].

It should be noted that a key ingredient to solve the SMV trajectory optimization problem is the ability to solve NLPs. While the general pseudospectral method and the hp-adaptive pseudospectral method [22], [24]–[27] can have a good performance approximating the problem, the efficiency depends largely on the initial guess value. If an initial guess, which is adequately close to the optimal solution, can be provided to the solver, then the efficiency of the NLP

solvers can be significantly improved and vice versa. For some optimal control software like GPOPS and DIRCOL, this process can usually be made by simply interpolating the boundary conditions provided by the user. However, due to the lack of physical knowledge, the initial guess value provided by the user may not be physically meaningful and it can hardly satisfy the dynamic model and path constraints. Therefore, the NLP solver will start at an infeasible point where most of the constraints are violated (i.e., cannot be satisfied) meaning the number of the Newton iteration will be increased.

To tackle this problem, this paper presents a combinatory approach that uses the violation learning differential evolution (VLDE) algorithm and the hp-adaptive pseudospectral method. The VLDE algorithm is presented as an initial guess generator. The fundamental framework of the proposed algorithm is based on [28]. Since the main difficulty for solving the SMV trajectory programming problem is the strict path constraints, a new constraint handling strategy using the violation and satisfactory degree [29], [30] is designed and embedded in the algorithm framework so that the feasibility of the calculated solution can be improved significantly. Compared with genetic algorithm (GA) [3], [31], differential evolution (DE) [31]–[33], particle swarm optimization (PSO), and other evolutionary algorithms, the proposed algorithm tends to be more stable at the first several iterations and has better convergence ability. Another advantage of the proposed algorithm is that the diversity of the population can be improved, so that it has the ability to avoid local optimal solution. In addition, since the proposed approach uses an adaptive way to crossover and mutate, it can offer the algorithm a higher speed to converge. Moreover, based on the definition of violation degree and satisfactory degree, a proper fitness function is designed that does not contain penalty factors or weight coefficients. Because of these advantages, the VLDE approach is applied to solve the SMV trajectory planning problem.

The paper is organized as follows. In Section II, the aerodynamic model of the SMV and some basic principles of the trajectory optimization problem are introduced. Section III describes the method used to discretize the optimal control problem. Then, in Section IV, the process of solving SMV trajectory problem using the VLDE algorithm is explained in detail. In Section V, a series of comparisons between the VLDE-based method and other methods are carried out.

II. PROBLEM DESCRIPTION

The overall mission can be seen in Fig. 1. The general skip re-entry can be divided into five phases: initial roll, down control, up control, Kepler, and final entry. Considering the mission of the SMV is to overfly the ground target with specific altitude, only the challenging phase 2 is taken into account in this paper.

A. SMV Dynamic Model

The Earth is considered as a symmetrical sphere and the Earth's rotation is ignored. The three degree-of-freedom

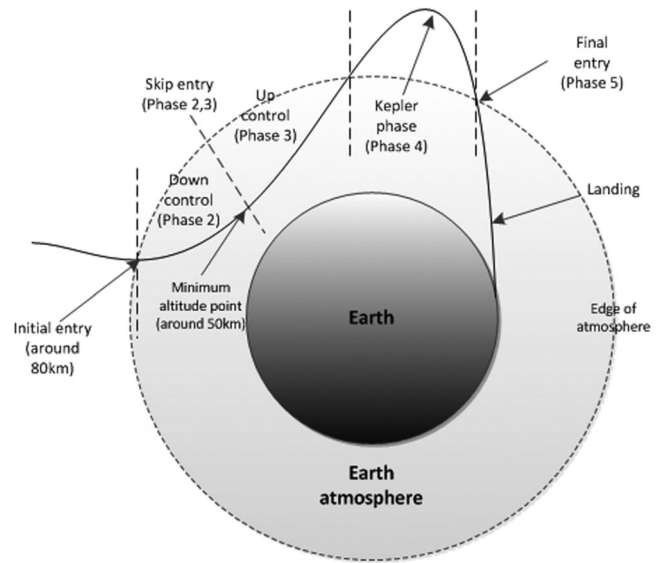


Fig. 1. Mission profile.

dynamic equations of the re-entry vehicle are given by (1). In the equations of motion, r is the radial distance from the Earth center to the vehicle, θ and ϕ are the longitude and latitude, respectively. V is the Earth-relative velocity, γ is the relative flight-path angle, ψ is the relative velocity heading angle measured clockwise from the north, m is the mass of vehicle, t is time. α and σ are angle of attack and bank angle, respectively. The state variables are $X = [r, \theta, \phi, V, \gamma, \psi, \alpha, \sigma]^T$, while the controls are α_c and σ_c . In reality, the actual control variables cannot change significantly (i.e., from lower bound to upper bound). Therefore, in the model provided in (1), two rate constraints are introduced, which can be concluded to the last two equations:

$$\begin{aligned}\dot{r} &= V \sin \gamma \\ \dot{\theta} &= \frac{V \cos \gamma \sin \psi}{r \cos \phi} \\ \dot{\phi} &= \frac{V \cos \gamma \cos \psi}{r} \\ \dot{V} &= -\frac{D}{m} - g \sin \gamma \\ \dot{\gamma} &= \frac{L \cos \sigma}{mV} + \left(\frac{V^2 - gr}{rV} \right) \cos \gamma \\ \dot{\psi} &= \frac{L \sin \sigma}{mV \cos \gamma} + \frac{V}{r} \cos \gamma \sin \psi \tan \phi \\ \dot{\alpha} &= K_\alpha (\alpha_c - \alpha) \\ \dot{\sigma} &= K_\sigma (\sigma_c - \sigma).\end{aligned}\tag{1}$$

The atmosphere model, lift L , and drag D can be formulated as

$$\begin{aligned}g &= \frac{\mu}{r^2} \quad \rho = \rho_0 \exp \frac{r - r_0}{h_s} \\ L &= \frac{1}{2} \rho V^2 C_L S D = \frac{1}{2} \rho V^2 C_D S\end{aligned}\tag{2}$$

where S is the reference area, ρ is the density of the atmosphere, and ρ_0 is the density of the atmosphere at sea level. r_0 is the Earth radius, C_L and C_D are lift and drag

coefficients determined by the angle of attack α and Mach number, respectively, and g is the gravity acceleration.

The drag and lift coefficients can be determined by the following equations:

$$\begin{aligned} C_D &= C_{D0} + C_{D1}\alpha + C_{D2}\alpha^2 \\ C_L &= C_{L0} + C_{L1}\alpha \end{aligned} \quad (3)$$

where C_{L0} , C_{L1} , C_{D0} , C_{D1} , and C_{D2} are mission-dependent lift and drag coefficients.

B. Re-Entry Process Constraints

- 1) Terminal state constraint: Due to the mission requirement, the descent phase should have strict boundary conditions. The terminal state constraints are given as

$$\begin{aligned} r(t_f) &= r_f \\ \gamma(t_f) &= \gamma_f \end{aligned} \quad (4)$$

where t_f is the terminal time of the mission and r_f , γ_f represent the altitude and flight path angle values at the final time.

- 2) Constraint of state and control variables: The tolerant region of state variables during the descent phase can be summarized as

$$\begin{aligned} r_{\min} &\leq r \leq r_{\max} & \theta_{\min} &\leq \theta \leq \theta_{\max} \\ \phi_{\min} &\leq \phi \leq \phi_{\max} & V_{\min} &\leq V \leq V_{\max} \\ \gamma_{\min} &\leq \gamma \leq \gamma_{\max} & \psi_{\min} &\leq \psi \leq \psi_{\max} \\ \alpha_{\min} &\leq \alpha \leq \alpha_{\max} & \sigma_{\min} &\leq \sigma \leq \sigma_{\max}. \end{aligned} \quad (5)$$

Similarly, the control variables should vary in its acceptable regions and can be written as

$$\begin{aligned} \alpha_{c(\min)} &\leq \alpha_c \leq \alpha_{c(\max)} \\ \sigma_{c(\min)} &\leq \sigma_c \leq \sigma_{c(\max)}. \end{aligned} \quad (6)$$

- 3) Path constraint: The heating rate, dynamic pressure, and normal acceleration constraints should be taken into account during the entire descent phase and it can be written as follows:

$$\begin{aligned} \dot{Q}_d &= K_Q \rho^{0.5} V^{3.07} (c_0 + c_1\alpha + c_2\alpha^2 + c_3\alpha^3) \\ g_1(t) &= \dot{Q}_d - \dot{Q}_{d\max} \leq 0 \\ P_d &= \frac{1}{2} \rho V^2 \\ g_2(t) &= P_d - P_{d\max} \leq 0 \\ n_L &= \frac{\sqrt{L^2 + D^2}}{mg} \\ g_3(t) &= n_L - n_{L\max} \leq 0 \end{aligned} \quad (7)$$

where $\dot{Q}_{d\max}$, $P_{d\max}$, and $n_{L\max}$ represents maximum allowable values for heating rate, dynamic pressure, and normal acceleration, respectively.

C. Objective Function of Trajectory Optimization

In this paper, to complete the mission in the shortest time minimizing the final time is chosen as the objective

function J :

$$J = \min t_f. \quad (8)$$

Therefore, the trajectory planning problem can be considered as an optimal control problem that has the minimum value of cost function and satisfies the terminal state constraints, state and control box constraints, three path constraints, and dynamics characterized in (1).

III. HP-ADAPTIVE PSEUDOSPECTRAL METHOD

To solve the SMV trajectory optimization problem using numerical methods, the continuous-time SMV trajectory optimization problem needs to be transformed to an NLP and the discretization method used in this paper is the hp-adaptive pseudospectral method. This method is an orthogonal collocation method, which the collocation points are the Legendre–Gauss–Radau (LGR) points [34]. The advantage of using the orthogonal collocation method over other methods is that the quadrature approximation to a definite integral is extremely accurate when the collocation points are chosen in an orthogonal manner.

Assume the time interval of an optimal control problem is $[t_0, t_f]$. Using the Radau pseudospectral method, mentioned in [22], the time interval should be transcribed via the affine transformation [34]. The LGR points are the root of linear combination of Legendre polynomials and it can be written as

$$P_{K-1}(\tau) + P_K(\tau) = 0 \quad (9)$$

where the K th-order Legendre polynomial $P_K(\tau)$ is given as

$$P_K(\tau) = \frac{1}{2^K K!} \frac{d^K}{d\tau^K} [(\tau^2 - 1)^K]. \quad (10)$$

Solving (9), the LGR points τ_i , $i = 1, \dots, N_k$, which is defined on $[-1, 1)$ (only including one of the endpoints) can be obtained. After generating the collocation points, the next step is to generate the approximation of the state and control variables. Let us consider N_k LGR collocation points $\tau_1, \dots, \tau_{N_k}$ on the interval $[-1, 1)$, with $\tau_1 = -1$ and $\tau_{N_k} < +1$. An additional point $\tau_{N_k+1} = 1$ is introduced, which is used to describe the approximation to the final state variables [34]. Then, in the Radau pseudospectral method [34], the state and control of optimal control problems are approximated as

$$\begin{aligned} x(\tau) &\approx X(\tau) = \sum_{i=1}^{N_k+1} X_i L_i(\tau) \\ u(\tau) &\approx U(\tau) = \sum_{i=1}^{N_k} U_i L_i(\tau) \end{aligned} \quad (11)$$

where $L_i(\tau)$, ($i = 1, \dots, N_{k+1}$) is the basis of lagrange polynomials. It is worth noting that the time domain for LGR points is $[-1, 1)$, while for Legendre–Gauss–Lobatto (LGL) points [25], this region becomes $[-1, 1]$. Since in the Legendre pseudospectral method, $\dot{P}_{K-1}(\tau) = 0$ is used

to calculate the LGL points. The differences between the LGR points and LGL points can be found in [22].

For an optimal control problem, which contains n_s state variables and n_c control variables, if the number of LGR collocation points is chosen as N_k , then the number of decision variables of the NLP solver is $n_s \times (N_k + 1) + (n_c \times N_k) + 1$ (as shown in (11), the index i for the state is from 1 to N_{k+1} , whereas the index for the control is from 1 to N_k , t_f is also an optimization parameter). In the hp strategy, if the mesh cannot meet the accuracy tolerance, then the number of nodes or interval should be changed either by increasing the degree of polynomial in the mesh interval or dividing the current mesh into several segments. By employing the hp strategy [20], the dynamic equation and path constraints can be transformed to

$$\sum_{j=1}^{N_k+1} D_{ij}^{(k)} X_j^{(k)} - \frac{t_k - t_{k-1}}{2} f(X_i^{(k)}; U_i^{(k)}; \tau_i^{(k)}; t_{k-1}; t_k) = 0$$

$$C(X_i^{(k)}, U_i^{(k)}, \tau_i^{(k)}; t_{k-1}, t_k) \leq 0 \quad (12)$$

where $k = 1, \dots, K$ and K is the number of mesh intervals. N_k is the number of LGR points and D is the differential matrix. f is the right-hand side of the model and $C = [g_1, g_2, g_3]^T$.

Let ε stand for an accuracy tolerance for the discrete algebraic function constraint. The error of the algebraic equation $a_s^{(k)}$ and path constraints $b_s^{(k)}$ at the s th collocation point and in the k th time interval can be calculated by

$$|\dot{X}^{(k)}(\tau_s^{(k)}) - \frac{t_k - t_{k-1}}{2} f(X_s^{(k)}, U_s^{(k)}, \tau_s^{(k)}; t_{k-1}, t_k)| = a_s^{(k)}$$

$$C(X_s^{(k)}, U_s^{(k)}, \tau_s^{(k)}; t_{k-1}, t_k) = b_s^{(k)}. \quad (13)$$

Therefore, the maximum error $e_{\max}^{(k)}$ in the k th time interval can be approximated as $e_{\max}^{(k)} = \max[a_s^{(k)}, b_s^{(k)}]$. If the equation $e_{\max}^{(k)} \leq \varepsilon$ can be satisfied, then the algorithm will stop the iteration because collocation points in the k th interval can reach the tolerance. Otherwise, it should be divided into subintervals or added more collocation points. Whether the current mesh gird should be divided into segments or added collocation points can be determined by using the curvature value. Suppose $k_{\max}^{(k)}$ and $\bar{k}^{(k)}$ are the maximum curvature and average curvature of all the nodes, respectively. Furthermore, let $r^{(k)} = \frac{k_{\max}^{(k)}}{\bar{k}^{(k)}}$ be the ratio of the maximum to the mean curvature. Setting the tolerance of curvature as r_{\max} and if $r^{(k)} \geq r_{\max}$, then the trajectory in this time interval tends to have oscillations and it should be divided into new subintervals. The number of the subinterval n_k is determined using $n_k = \text{ceil}(\log(e_{\max}^{(k)}/\varepsilon))$, where the function of $\text{ceil}(\cdot)$ is to round a number to the next larger integer. On the other hand, if the tolerance can satisfy $r^{(k)} < r_{\max}$, the trajectory tends to be flat in this time interval and the accuracy can be improved by adding more collocation points. The number of points that should be added is determined using $N_k = N_k + \text{ceil}(\log(e_{\max}^{(k)}/\varepsilon))$.

IV. TRAJECTORY OPTIMIZATION PROBLEM BASED ON THE VLDE METHOD

Based on the method presented in Section III, the trajectory optimization problem is transformed to an NLP problem. There are many well-developed NLP algorithms and solvers. However, most of the existing methods tend to be sensitive with regards to the initial guess and if the solver receives an initial guess that is close to the optimal solution, the efficiency can be improved significantly. Therefore, it is essential to design an initial guess generator. In this paper, a VLDE method was designed. The structure of this combinational method can be summarized as follows. First, the initial guess for the NLP problem is generated using the VLDE method and after getting the initial guess, a gradient-based optimization method is applied to find the optimal solution.

DE is a simple, efficient, and robust algorithm. It has been widely used in a number of scientific and engineering fields as a global optimization technique. There are four main procedures for the DE algorithm: initialization, selection, crossover, and mutation. In the initialization part, after the scale of the population and the number of iteration times are assigned, the first population needs to be generated. According to the procedure described in (9)–(11), the optimization parameters, angle of attack, and bank angle should be chosen randomly at the collocation points satisfying the control constraints given by (6). This can be written as

$$\text{pop}_{ij,G} = U_{l_j} + \text{rand} \times (U_{h_j} - U_{l_j}), (i = 1, 2 \dots \text{NP}; j = 1, 2 \dots N_k) \quad (14)$$

where $\text{pop}_{ij,G}$ is each chromosome for the G th generation, U_{l_j} is the minimum value of the control variables, whereas U_{h_j} is the maximum value of control variables. $\text{rand} \in [0, 1]$ is a random number. NP is the size of population. Since the aim of the initial trajectory generator is to provide a reference trajectory to the NLP solver, to reduce the computational burden of the initial guess generator, only the control variables are parameterized based on (11). The time history of the state variables can then be calculated by integrating the equations of motion with a fourth-order Runge–Kutta method. Before introducing the VLDE algorithm, the violation degree of the constraints should be defined.

A. Violation Degree of Constraints

Generally, for NLP problems, there are two kinds of constraints, equality constraints and inequality constraints. In terms of inequality constraints, the violation degree for the relation “ \leq ” (i.e., $g(x) \leq g_i^*$) can be defined as

$$\mu_{g_{ij}}(x, u) = \begin{cases} 0, & g_{ij}(x, u) \leq g_i^*; \\ \frac{g_{ij}(x, u) - g_i^*}{g_{ij}^{\max} - g_i^*}, & g_i^* \leq g_{ij}(x, u) \leq g_{ij}^{\max}; \\ 1, & g_{ij}(x, u) \geq g_{ij}^{\max} \end{cases} \quad (15)$$

where g_{ij} is the value of constraints for each individual at each node, while the tolerance range should be (g_i^*, g_{ij}^{\max}) . The criterion for selecting g_i^* is based on the upper bound of the corresponding path constraints (e.g., the allowable maximum heating rate, dynamic pressure, and load factor). Take heating constraint as an example, g_i^* can be set to 200 BTU. Since each individual in the current population represents a trajectory, g_{ij}^{\max} stands for the maximum value of the constraint for the i th individual during the whole time history. If the individual can satisfy the constraint described in (7), then the corresponding violation degree should equal to 0. If not, then the violation degree should be calculated based on (15).

Similarly, for relation “ \geq ” (i.e., $g(x) \geq g_i^*$), the violation function can be defined as

$$\mu_{g_{ij}}(x, u) = \begin{cases} 0, & g_{ij}(x, u) \geq g_i^*; \\ \frac{g_i^* - g_{ij}(x, u)}{g_i^* - g_{ij}^{\max}}, & g_{ij}^{\min} \leq g_{ij}(x, u) \leq g_i^*; \\ 1, & g_{ij}(x, u) \leq g_{ij}^{\min} \end{cases} \quad (16)$$

where (g_{ij}^{\min}, g_i^*) can be treated as the tolerance region.

For equality constraints (i.e., $g(x) = g_i^*$), the tolerance region $(g_{ij}^{\min}, g_{ij}^{\max})$ can be chosen based on decision maker's preference or by using the minimum and maximum constraint values at all of the nodes. Its violation function is given as

$$\mu_{g_{ij}}(x, u) = \begin{cases} 1, & g_{ij}(x, u) \geq g_{ij}^{\max}; \\ \frac{g_{ij}(x, u) - g_i^*}{g_{ij}^{\max} - g_i^*}, & g_i^* \leq g_{ij}(x, u) \leq g_{ij}^{\max}; \\ 0, & g_{ij}(x, u) = g_i^*; \\ \frac{g_i^* - g_{ij}(x, u)}{g_i^* - g_{ij}^{\min}}, & g_{ij}^{\min} \leq g_{ij}(x, u) \leq g_i^*; \\ 1, & g_{ij}(x, u) \leq g_{ij}^{\min}. \end{cases} \quad (17)$$

In this way, the value of the violation function can directly reflect the magnitude of the solution violating the constraints and because of $\mu \in [0, 1]$, the overall violation degree for each individual can be calculated by

$$V_i = \sum_{j=1}^{N_k} \frac{\mu_{g_{ij}}(x, u)}{N_k}, \quad i = 1, 2, \dots, \text{NP}. \quad (18)$$

On the other hand, it should be noted that the violation degree can also be used to reflect the satisfactory degree of individuals by introducing $S_i = 1 - V_i$. Since both S_i and V_i are designed to reflect the preference of solutions with respect to constraints, another satisfactory degree $\mu_{J_i}(x, u)$ should be introduced to reflect the preference of individuals with respect to objective. In terms of minimizing problems, for instance, its objective satisfactory degree can be

computed as

$$\mu_{J_i}(x, u) = \begin{cases} 1, & J_i(x, u) \leq J^*; \\ 1 - \frac{J_i(x, u) - J^*}{J_i^{\max} - J^*}, & J^* \leq J_i(x, u) \leq J_i^{\max}; \\ 0, & J_i(x, u) \geq J_i^{\max} \end{cases} \quad (19)$$

where J_i^{\max} is the maximum value of J in the current population, while J^* can be an ideal value or a rough value decided by decision makers.

B. Fitness Function With Violation Degree

Classical evolutionary-based algorithms usually use a penalty function to create the augmented fitness function. However, the optimization process may largely depend on the penalty parameters, and because of this, the penalty factors should be chosen in a proper way. Motivated by this argument, the following fitness function term is developed, where the infeasibility is taken into account based on violation degree:

$$F_i = \frac{f_i + 1}{2}, \quad i = 1, 2, \dots, \text{NP} \quad (20)$$

where F_i is the fitness value of the i th individual and f_i is calculated by

$$f_i = \begin{cases} \mu_{J_i}, & \text{if } S_i = 1; \\ \mu_{J_i} - \max(V_i), & \text{if } S_i \neq 1. \end{cases} \quad (21)$$

Equation (20) is taken as the fitness value in the VLDE method, whose theoretic values are in the range $[0, 1]$.

C. Evolutionary Strategies

There are three evolutionary procedures in the proposed method: selection, direct search, and adaptive mutation. In the selection process, to create the next generation, individuals who have high value of fitness function are selected to the mating pool until the number of individuals reaches a certain number. It is described as

$$\text{pop}_{i,G+1} = \begin{cases} \text{pop}_{i,G}, & \exp\left(\frac{F_i - F_j}{G + 1}\right) > \text{rand}; \\ \text{pop}_{j,G}, & \exp\left(\frac{F_i - F_j}{G + 1}\right) < \text{rand}. \end{cases} \quad (22)$$

As can be seen from (22), not all the candidates with high fitness can be selected into the next generation. The benefit of using such a strategy is to keep the variety of population and get global solution.

In the normal crossover strategy, the new candidates generated by applying crossover function may not have improvement compared with the previous generation. Therefore, to improve the convergence speed, procedures embedded in the Nelder–Mead direct search algorithm [35] are applied to get the next generation. The method uses the concept of simplex and three candidates are chosen to project on a plane (working simplex S). General processes for the Nelder–Mead direct search are summarized as follows.

- 1) Ordering and centroid: Based on the fitness value calculated by (20), the worst $F_h = F_{\text{pop}_{h,G}}$, the second worst $F_s = F_{\text{pop}_{s,G}}$, and the best $F_l = F_{\text{pop}_{l,G}}$ candidates can be found in the current working simplex. Then, the centroid c of the best sides is obtained using

$$c = \frac{1}{2}(\text{pop}_{s,G} + \text{pop}_{l,G}). \quad (23)$$

- 2) Reflection: The reflected point is obtained by

$$\text{pop}_{r,G+1} = c + \omega_1(c - \text{pop}_{h,G}). \quad (24)$$

If the fitness value of reflected point is better than the second worst and no better than the best (i.e., $F_s < F_r \leq F_l$), then keep $\text{pop}_{r,G+1}$ to the next generation.

- 3) Expansion: If the reflected point is better than the best point (i.e., $F_r > F_l$), then a greedy searching can be employed to get an expanded point $\text{pop}_{e,G+1}$ described by

$$\text{pop}_{e,G+1} = \text{pop}_{r,G+1} + \omega_2(\text{pop}_{r,G+1} - c). \quad (25)$$

If the expanded point has a better performance than the reflected point, then keep the expanded point to the next generation. Otherwise, keep the reflected point to the next generation.

- 4) Contraction: If the reflected point is no better than the second worst candidate (i.e., $F_r \leq F_s$), the contracted point is calculated by

$$\text{pop}_{c,G+1} = c + \omega_3(\text{pop}_{h,G+1} - c). \quad (26)$$

If the contracted point is better than the worst in terms of fitness value, then keep the contracted point to the next generation.

- 5) Shrink transformation: If the contracted point is still not better than the worst point, then all the points are replaced with

$$\text{pop}_{i,G} = \text{pop}_{l,G} + \omega_4(\text{pop}_{i,G} - \text{pop}_{l,G}), \quad i = 1, 2 \quad (27)$$

and go to step 1).

The $\omega_1 = 1$, $\omega_2 = 2$, $\omega_3 = 0.5$, and $\omega_4 = 0.5$ are the reflection, expansion, contraction, and shrink transformation coefficients, respectively. Compared with the traditional crossover function, this strategy can ensure the next generation is not worse than the previous generation and improve the convergence speed.

Mutation operation is used to avoid local optimal solution and usually p_i and p_{ij} are chosen as mutation rate and mutation rate of segment, respectively. In VLDE, to improve the speed of convergence, p_i and p_{ij} are set as

$$p_i = V_i, \quad p_{ij} = \mu_{g_{ij}}. \quad (28)$$

In the process described in (28), individuals who have high value of constraint violation will gain a higher probability to mutate at those high violation segments. Clearly, these two values will change adaptively with the increasing of generation.

D. Learning Strategy

Commonly, in the selection part, since it only chooses solutions that have a competitive performance in terms of fitness value at the current generation and eliminates bad solutions, the diversity of the population is lost. If a solution at the current generation has a bad performance in terms of fitness value but after running the evolutionary procedures, it can be improved massively, then the offspring should also be kept to the next generation. This improvement degree can be treated as a solution's learning ability and can be defined by

$$P_{i,G+1} = S_{i,G+1} - \frac{S_{i,G} + S_{j,G} + S_{k,G}}{3} \quad (29)$$

where $S_{i,G}$, $S_{j,G}$, and $S_{k,G}$ are the satisfactory degree of three candidates who are not selected in the selection part and $S_{i,G+1}$ is the offspring calculated by applying the evolutionary procedures. A high value of P means the candidate is on an evolutionary direction which can achieve a massive improvement. Consequently, the offspring are made up of the following three parts:

- 1) individuals who have high value of fitness function in the G th generation;
- 2) the offsprings of the selected parents;
- 3) individuals who tend to have a massive improvement compared to the G th generation.

E. Optimization by the VLDE Algorithm

Considering the whole VLDE approach, the optimization procedures can be summarized as follows:

- 1) randomly generate initial candidates and calculate the violation degree, fitness value based on (15)–(21);
- 2) apply the evolutionary procedures to obtain new candidates for next generation;
- 3) Use learning strategy and perform step 1)–3) repeatedly until reaching the maximum iteration time.

By using the proposed algorithm, the initial guess for the SMV trajectory optimization problem can be calculated. The flow chart of the VLDE method is presented in Fig. 2.

V. SIMULATION RESULTS

A. Simulation Parameter Setting

To invest the feasibility and optimality of the proposed method, a number of simulation experiments were carried out. Comparative studies with other global optimization methods such as DE, PSO, GA [36], and artificial bee colony (ABC) [36] are presented. All the simulations carried out in this paper are under Windows 7 and Intel(R) i7-3520M CPU, 2.90 GHZ, with 4.00 GB RAM. In this section, all the values of state and control variables at the initial and final time are listed in Table I, and the control parameters of global methods are given in Table II.

The vehicle-dependent and mission-dependent parameters are set as $S = 2690 \text{ ft}^2$, $\rho_0 = 0.002378 \text{ slug/ft}^3$, $C_{L0} = -0.2070$, $C_{L1} = 1.676$, $C_{D0} = 0.07854$, $C_{D1} = -0.3529$,

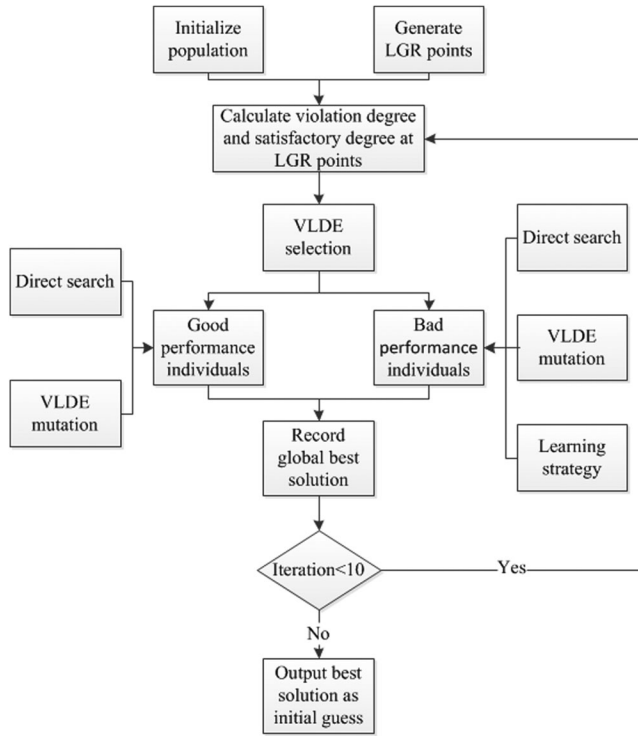


Fig. 2. Flow chart of the VLDE approach.

TABLE I
Simulation Parameter Setting

	Initial Value	Final Value	Minimum	Maximum
Altitude (ft)	21 162 900	21 066 900	21 066 900	21 162 900
Velocity (ft/s)	25 600	free	2000	45 000
Longitude (°)	0	free	−180	180
Latitude (°)	0	free	−70	70
Fpa (°)	−1	0	−80	80
Heading angle (°)	90	free	−180	180
AOA (°)	17.43	free	−10	30
Bank angle (°)	−75	free	−90	1
Mass (slug)	6309.43	6309.43	6309.43	6309.43

TABLE II
Control Parameter for Global Algorithms

VLDE		DE		PSO		GA		ABC	
NP	50	NP	50	NP	50	NP	50	NP	50
Iter	10	Iter	10	Iter	10	Iter	10	Iter	10
ω_1	1	CR	0.5	w_{\max}	0.8	p_i	0.2	limit	10
ω_2	2	p_i	0.2	w_{\min}	0.2	p_{ij}	0.5		
ω_3	0.5	p_{ij}	0.5	c_1	2	CR	0.5		
ω_4	0.5			c_2	2				

$C_{D2} = 2.040$, $\dot{Q}_{d\max} = 200 \text{ BTU}/(\text{s} \cdot \text{ft}^2)$, $P_{d\max} = 13406.4583 \text{ Pa}$, $n_{L\max} = 2.5$, and $N_k = 25$, respectively. The specific boundary conditions are given by the industrial sponsor and only the descent phase shown in Fig. 1 is taken into account in this paper. The initial altitude is around 80 km, where the edge of the atmosphere is assumed to start at.

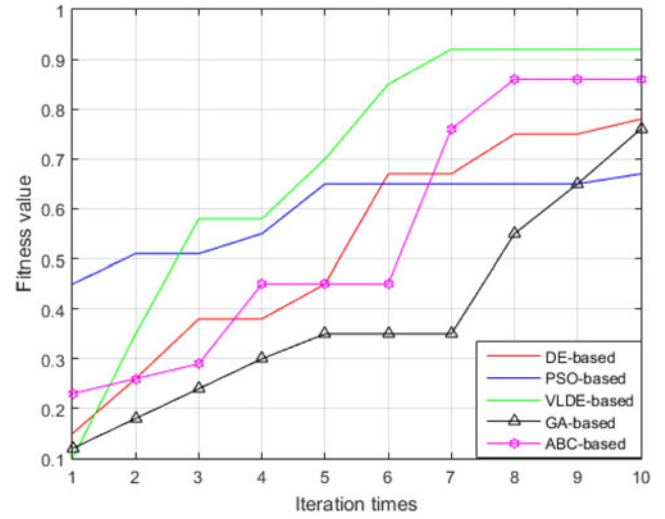


Fig. 3. Scenario 1.

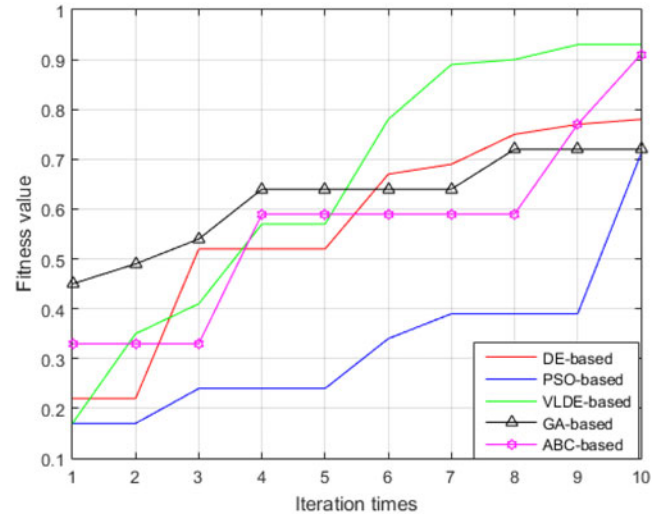


Fig. 4. Scenario 2.

B. Numerical Simulation

Based on the control parameters given in Table II, a series of simulations in terms of the convergence analysis for VLDE is conducted by testing three different final altitude scenarios. Specifically, the final altitude values are set as $r_1(t_f) = 21066900$, $r_2(t_f) = 21074900$, and $r_3(t_f) = 21080900$ for each scenario. Figs. 3–5 illustrate the results of the proposed VLDE algorithm and other global optimization methods under the same computational efforts. The results is projected onto the fitness value versus iteration time plane.

As can be seen from Figs. 3–5, the fitness value of the VLDE approach can finally converge to around 1 for each scenario within less iterations and the final values of the fitness function for VLDE algorithm are always higher than its counterparts. Therefore, although compared with the DE, PSO, GA, and ABC, the complexity of the proposed VLDE approach is higher, it can always perform a better convergence ability and quicker convergence speed, which means

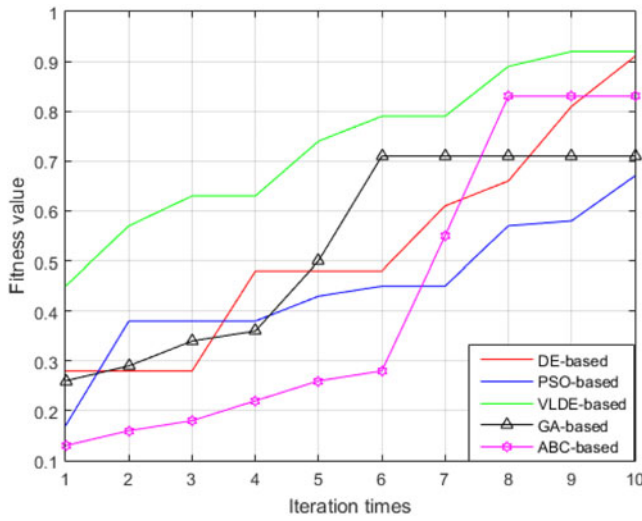


Fig. 5. Scenario 3.

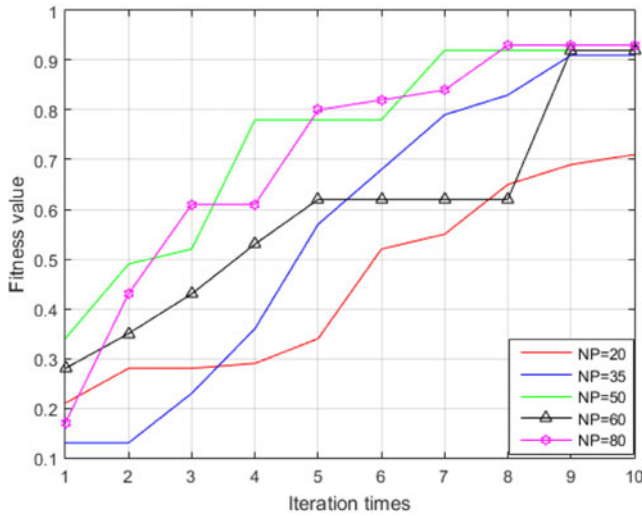


Fig. 6. Different number of population.

by using limited computational efforts, the quality of the initial guess generated using the VLDE algorithm is better than other methods investigated in this paper. Moreover, in reality, due to the computational burden should be taken into account and because of this, the iteration time is set as 10 for the VLDE algorithm.

To set a reasonable number of population NP, another simulation is carried out by fixing the number of iteration as 10. The fitness values of different numbers of NP are shown in Fig. 6. The results show that a larger number of NP can provide a higher final fitness value. However, for the initial guess generator, the aim is to provide a good reference trajectory to the NLP solver with less computational effort. Based on the simulation results shown in Fig. 6, in order to keep a balance between the quality of initial guess and the computational burden, NP is chosen as 50. Based on the control parameters given in Table II, the initial trajectories generated using different algorithms are plotted in Figs. 7 and 8.

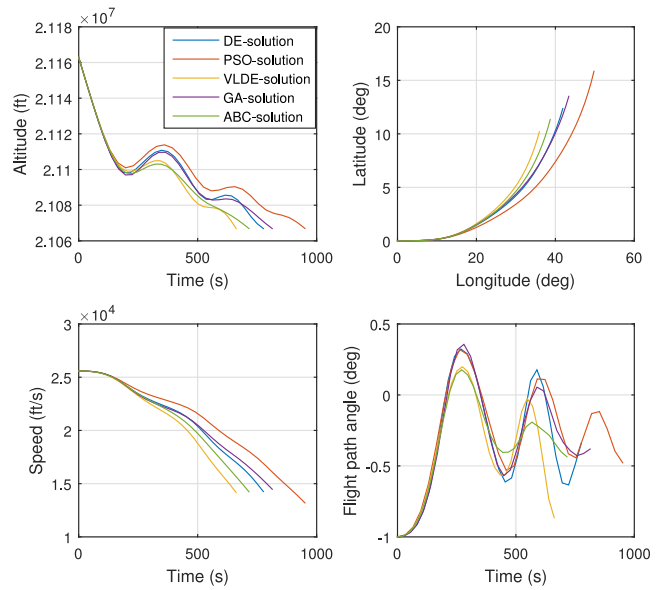


Fig. 7. Solution of different initial guess generators.

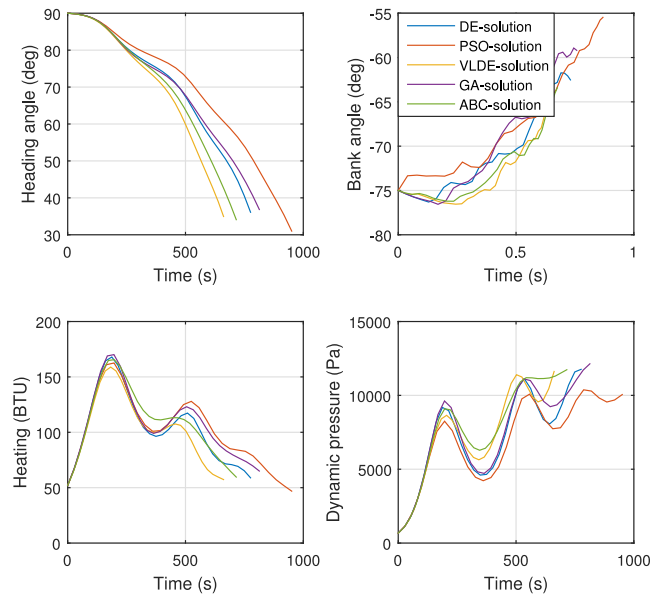


Fig. 8. Solution of different initial guess generators.

After calculating the solution from VLDE, the next step is to use the solution shown in Figs. 7 and 8 as an initial guess and provide it to the gradient-based NLP solver (SQP method). A comparison among the VLDE-based method, DE-based method, PSO-based method, GA-based method, and ABC-based method is presented to investigate the feasibility of the five combinational strategies. By setting $\epsilon = 1 \times 10^{-6}$, Figs. 9 and 10 show the time history of all the states and controls.

Table III illustrates the results of the four methods in terms of the computational time, Newton iteration time and values of the cost function. Generally, if the initial guess is not close enough to the optimal solution, the performance of gradient-based method is time consuming and it has high probability to converge to a local optima. From Figs. 9 and

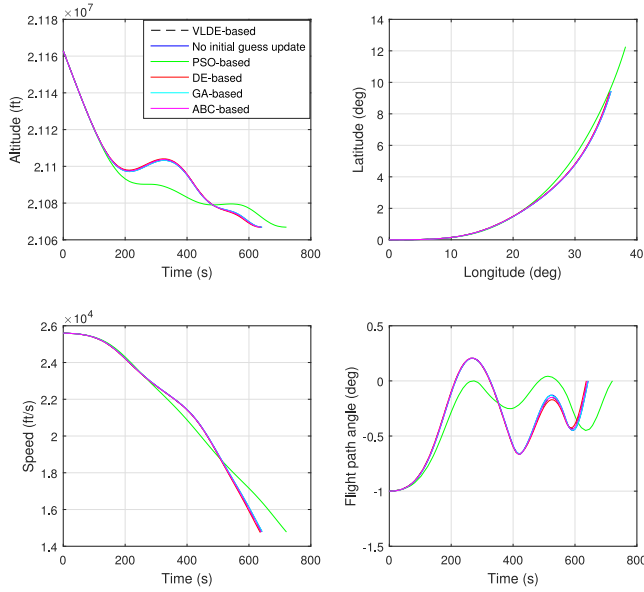


Fig. 9. States and controls generated by using six methods for Scenario 1.

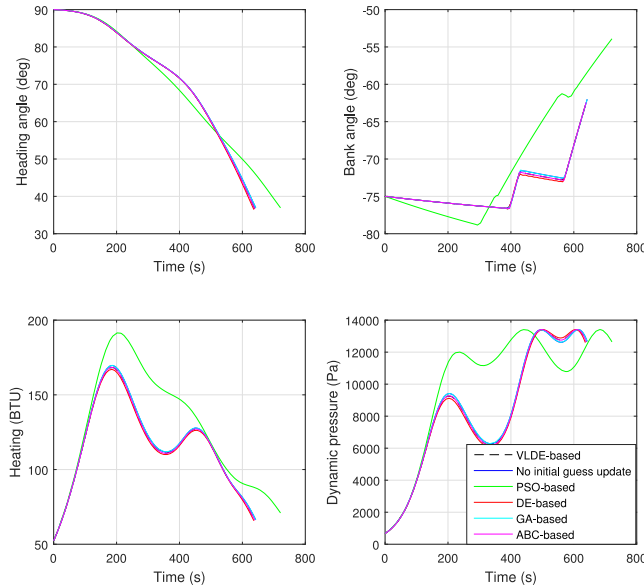


Fig. 10. States and controls generated by using six methods for Scenario 1.

10, the results are significantly different between the PSO-based method and others. This can be explained by the fact that in this case, the initial guess generated by the PSO-based method is not close enough to the optimal solution. This can be proved partly based on Figs. 3–5. Therefore, the reference trajectory generated using the PSO algorithm may close to a local optimum. When the NLP solver receives an infeasible or bad quality initial guess, the solver may spend a large number of Newton iterations and can hardly jump out from a local minimum convergence region. In addition, according to the results illustrated in Figs. 9 and 10, the trajectory obtained by VLDE-based methods can satisfy the path constraints in the entire mission. Therefore, the structural and thermal safety of the SMV is guaranteed, which is

TABLE III
Performance for Different Methods

Scenario 1	CPU Time for NLP Solver (s)	NLP Solver Iter	J (s)
No initial update	22.31	3278	642.54
DE based	11.85	462	640.43
PSO based	20.44	2565	721.23
VLDE based	7.26	347	639.52
GA based	12.22	604	642.48
ABC based	9.53	448	639.52

Scenario 2	CPU Time for NLP Solver (s)	NLP Solver Iter	J (s)
No initial update	55.05	7489	809.06
DE based	28.17	686	795.22
PSO based	30.23	837	795.45
VLDE based	21.72	585	793.38
GA based	29.85	703	796.23
ABC based	26.48	654	796.08

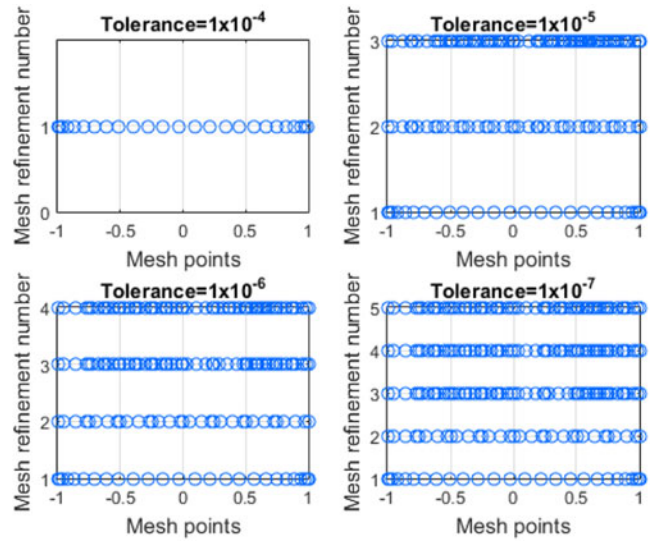


Fig. 11. Mesh grid distribution with different accuracy tolerances.

the prerequisite for the validity of an approach to trajectory optimization. Moreover, in order to detail the influence of the hp strategy on the proposed algorithm, the distribution of time nodes with different predefined accuracy tolerances ϵ is shown in Fig. 11. The ways in which this strategy adds more points in a specific time interval or divides an interval into subintervals follow the discussion followed in Section III. Using the hp strategy, the time history for the states and controls can be much smoother. This can be seen from Figs. 9 and 10, where the distribution of grid points tends to be dense at those areas having a high value of curvature, while the distribution of discrete points tends to be sparse at flat areas.

To further analyze the performance of the hp-pseudospectral method with the proposed initial guess generator, another mission with most of final states constrained and smaller limits on the path constraints is carried out. In this case, the final boundary conditions are $[r(t_f), \phi(t_f), V(t_f), \gamma(t_f)]^T = [21066900 \text{ ft}, 11.80^\circ, 14000 \text{ ft/s}, 0^\circ]^T$. The maximum allowable heating, dynamic

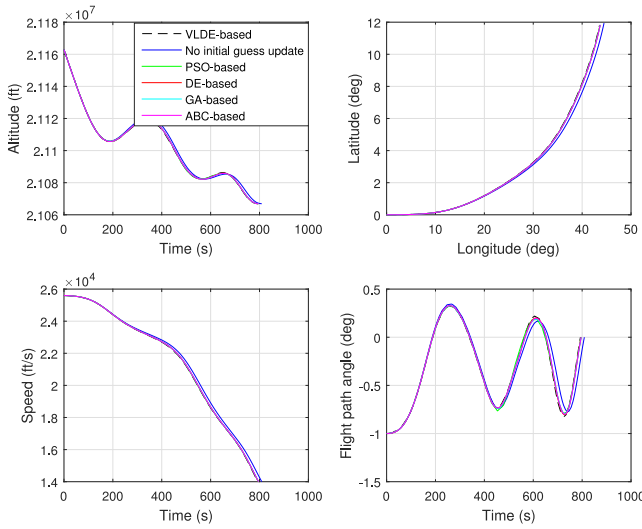


Fig. 12. States and controls generated by using six methods for Scenario 2.

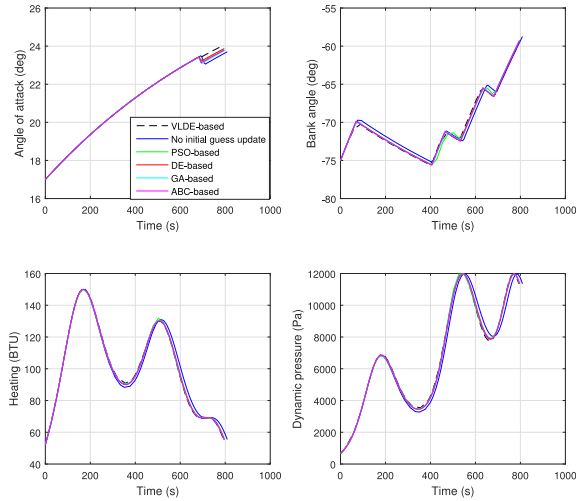


Fig. 13. States and controls generated by using six methods for Scenario 2.

pressure, and load factor are set as: $[Q_{dmax}, P_{dmax}, n_{Lmax}] = [150 \text{ BTU}, 11970.0521 \text{ Pa}, 2.5]$. Since the constraints become hard to satisfy, the number of iterations for the initial guess generator is set as $\text{Iter} = 20$. The time history of the state and control variables is shown in Figs. 12 and 13. Detailed results in terms of number of NLP solver iteration (Newton iteration), computational time, and objective function values can be found in Table III (Scenario 2 part).

Although VLDE-based, DE-based, PSO-based, GA-based, and ABC-based methods managed to generate SMV trajectory between the initial position and terminal position, the data in Table III reveal that VLDE-based method can generate an optimal solution, and perform a smaller value of computational time and number of Newton iterations than others. Especially, compared with method without initial guess update, the number of NLP solver processing time (Newton iteration) and computational time can be saved significantly. It is worth noting that for some optimal control solvers, since there is no initial guess update process embedded in the software framework, the solver simply

interpolates the state and control variables using the boundary conditions provided by the user and uses it as the start point of the Newton iteration. In this way, the initial trajectory can hardly satisfy the dynamic equations and path constraints, which means most of the constraints will be violated and the number of Newton iteration tends to increase significantly. This can be reflected in Table III, where the number of NLP iteration for no initial guess update case is generally higher than other cases.

C. Analysis of the Simulation Result

To judge the quality of the solution generated by the VLDE-based algorithm, the first-order necessary conditions for an optimal control problem should be used. The dynamic model described in (1) is abbreviated as $\dot{x} = f(x, u)$. Then, the Hamiltonian function is constructed as

$$H = \lambda^T(t) f(x, u) \quad (30)$$

in which $\lambda(t) = [\lambda_r, \lambda_\theta, \lambda_\phi, \lambda_v, \lambda_\gamma, \lambda_\psi, \lambda_\alpha, \lambda_\sigma]^T$ is the costate variable corresponding to the state variables. The continuous-time first-order necessary conditions in terms of costate and Hamiltonian can be shown as

$$\begin{aligned} \lambda(\tau_0) &= -\frac{\partial J}{\partial x(\tau_0)} + v^T \frac{\partial \phi}{\partial x(\tau_0)}, \quad \lambda(\tau_f) = -\frac{\partial J}{\partial x(\tau_f)} + v^T \frac{\partial \phi}{\partial x(\tau_f)} \\ H(t_0) &= \frac{\partial J}{\partial t_0} - v^T \frac{\partial \phi}{\partial t_0}, \quad H(t_f) = \frac{\partial J}{\partial t_f} - v^T \frac{\partial \phi}{\partial t_f} \end{aligned} \quad (31)$$

where ϕ is boundary condition illustrated in Table I, and v is the Lagrange multiplier associated with the boundary condition. Based on (31), the costate value for each state variable at the final time should hold (32), which means if the state variables are fixed at the initial point, then for those states which have free final conditions, the estimated costate should be equal to 0:

$$\begin{cases} \lambda_\theta(t_f) = \frac{\partial J}{\partial \theta} \big|_{t=t_f} = 0 \\ \lambda_\phi(t_f) = \frac{\partial J}{\partial \phi} \big|_{t=t_f} = 0 \\ \lambda_v(t_f) = \frac{\partial J}{\partial v} \big|_{t=t_f} = 0 \\ \lambda_\psi(t_f) = \frac{\partial J}{\partial \psi} \big|_{t=t_f} = 0 \\ \lambda_\alpha(t_f) = \frac{\partial J}{\partial \alpha} \big|_{t=t_f} = 0 \\ \lambda_\sigma(t_f) = \frac{\partial J}{\partial \sigma} \big|_{t=t_f} = 0. \end{cases} \quad (32)$$

Moreover, to calculate the final value of the Hamiltonian function, the endpoint Lagrangian is formulated as

$$\begin{aligned} H(X(t_f), t_f, v) &= -\frac{\partial J}{\partial t_f} + v_r(r_f - r(t_f)) + v_\gamma(\gamma_f - \gamma(t_f)) \\ &= -1. \end{aligned} \quad (33)$$

Equation (33) indicates that the final value of the Hamiltonian should be zero for this problem. Then, the Hamiltonian

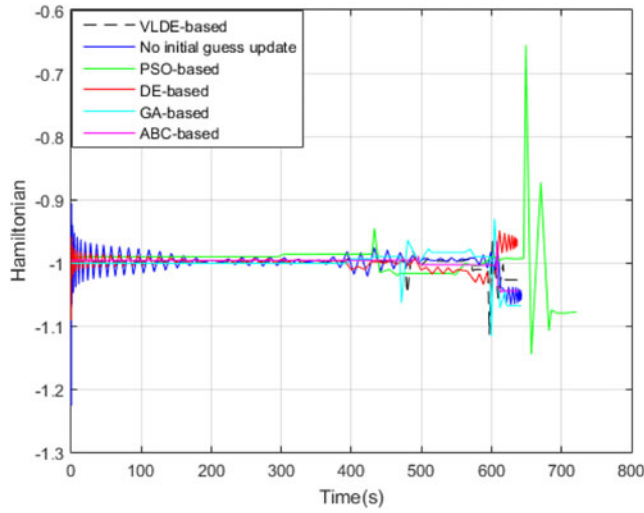


Fig. 14. Hamiltonian values for mission scenario 1.

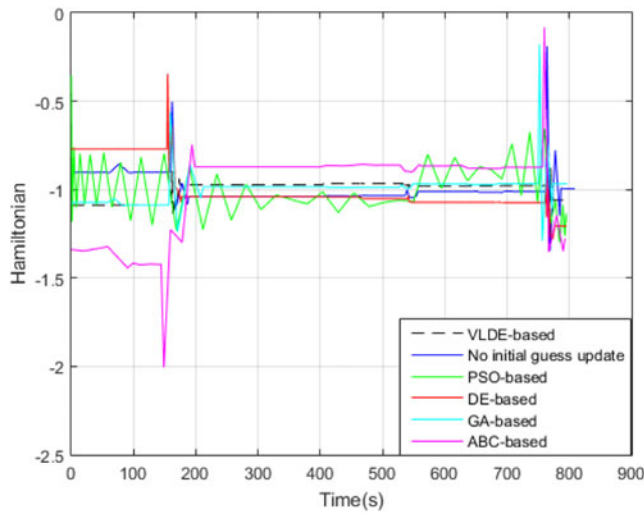


Fig. 15. Hamiltonian values for mission scenario 2.

evolution equation is used to demonstrate the behavior of the Hamiltonian with respect to time such that

$$\frac{\partial H}{\partial t} = 0. \quad (34)$$

Combining (34) and (33), it is clear that the following equation should be satisfied:

$$H(x^*(t_f), u^*(t_f), t_f) = -1, \quad H = \text{constant}. \quad (35)$$

Figs. 14 and 15 illustrate the value of Hamiltonian function calculated using the proposed method for the two mission scenarios described in Table III, whereas Table IV shows the final value of costate variables obtained using the VLDE approach.

Figs. 14 and 15 and Table IV demonstrate that compared with other methods, the VLDE-based method tends to approximate the Hamiltonian more accurately. Specifically, the value of Hamiltonian generated from the VLDE-based approach tends to keep flat around -1 with less variance. In addition, the final costate values for states that have free

TABLE IV
Final Costate Value Calculate by VLDE-Based Method

States	r	θ	ϕ	V
Initial point	Fixed	Fixed	Fixed	Fixed
Final point	Fixed	Free	Free	Free
λ_f	—	0	0	6.40×10^{-18}
States	γ	ψ	α	σ
Initial point	Fixed	Fixed	Fixed	Fixed
Final point	Fixed	Free	Free	Free
λ_f	—	0	2.40×10^{-13}	1.22×10^{-13}

TABLE V
Dispersions in the Entry Initial Conditions and Other Parameters

State/Parameter	Distribution	3- σ Value/Range
Altitude, ft	Zero-mean Gaussian	500
Longitude, $^\circ$	Zero-mean Gaussian	0.0749
Latitude, $^\circ$	Zero-mean Gaussian	0.3202
Velocity, ft/s	Zero-mean Gaussian	100
Flight-path angle, $^\circ$	Zero-mean Gaussian	0.1484
Heading angle, $^\circ$	Zero-mean Gaussian	0.0973
Vehicle's mass, slug	Uniform	$\pm 5\%$

final conditions are equal to 0 or approximately 0. That means the VLDE-based method can have a better performance in terms of following the original theory and it further confirms that the method proposed is feasible and effective for handling SMV trajectory optimization problem.

D. Dispersion Models

The purpose of dispersion simulations is to demonstrate the efficiency and feasibility of the proposed algorithm in the presence of significant deviations in trajectory state variables and vehicle uncertainties. The dispersions in terms of the initial entry conditions were modeled by zero-mean Gaussian dispersions. Furthermore, the vehicle mass was perturbed uniformly up to 5% with the nominal mass of 6309.43 slug, this gives a range of value of 5993.96 to 6624.90 slug. These entry-condition dispersions are the same with the data reported in the literature [37]. All the random initialization data used in the simulation are tabulated in Table V.

Based on the data given in Table V, Monte Carlo simulations were carried out and the results are tabulated in Table VI. For the mission described in Table I, 500 Monte Carlo simulations were performed. The time histories of 100 dispersed trajectories for the state and control variables calculated using the proposed algorithm are plotted in Figs. 16 and 17. Correspondingly, Table VI summarizes the statistic performance of the proposed algorithm in terms of the computational time and number of Newton iteration for 500 dispersed trajectories. It can be seen from Figs. 16 and 17 that the performance of the proposed method can be accepted, since it can successfully converge to the optimal solution and it is not sensitive with respect to the random initialization. Based on the data provided in Table VI, the performance of the proposed approach is generally better than other initial guess generators considered in this paper

TABLE VI
Statistics on the Algorithm Performance in 500 Dispersed Trajectories

Computational Time (s)	VLDE Based	DE Based	PSO Based	GA based	ABC Based
Average	8.69	12.61	15.96	12.04	10.58
Maximum	12.25	16.33	23.17	14.42	13.09
Minimum	7.03	10.55	11.23	10.28	9.17
Newton Iteration	VLDE Based	DE Based	PSO Based	GA Based	ABC Based
Average	338	503	741	563	447
Maximum	591	747	2662	783	644
Minimum	147	385	513	394	329

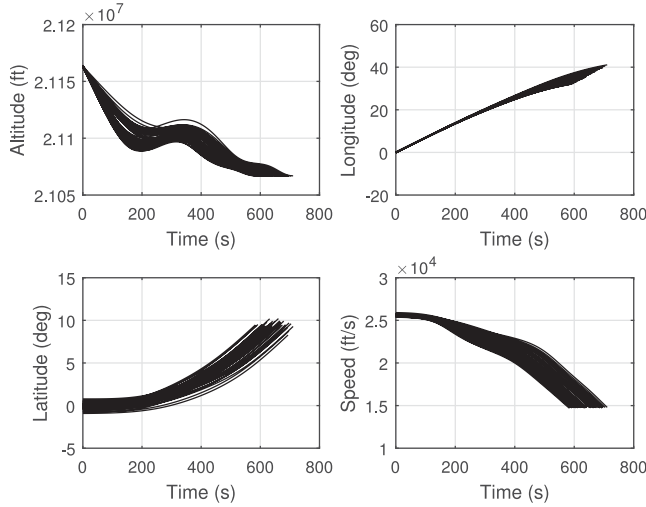


Fig. 16. Hundred dispersed trajectories for the state and control variables.

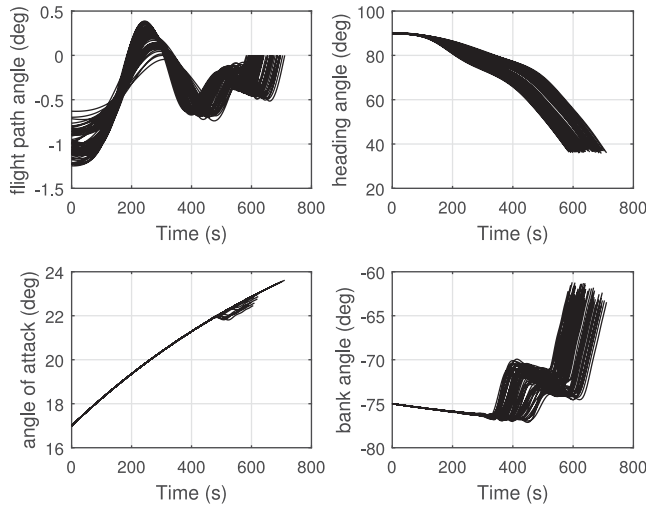


Fig. 17. Hundred dispersed trajectories for the state and control variables.

in terms of the computational time and numbers of NLP solver iterations. This indicates that the method designed in this study can have better performance over other evolutionary initial guess generators.

In summary, all the figures and data provided earlier confirm the feasibility and optimality of the proposed

VLDE-based method. By using the initial guess calculated using the VLDE algorithm, the SMV re-entry vehicle can reach the target position without violating the path constraints and boundary conditions. Moreover, it was shown that the VLDE algorithm has better convergence ability and convergence speed than its counterparts, DE, PSO, GA, and ABC algorithms in the current problem, which means it can have a better performance under low computational resources circumstance.

VI. CONCLUSION

In this paper, the application of the hp-adaptive discrete method to three dimensional SMV re-entry trajectory optimization problem has been introduced. In order to tackle with the sensitivity in terms of the initial guess for current gradient-based NLP solvers, a VLDE-based algorithm is designed. To avoid the design of penalty factors, the violation degree and satisfactory degree are introduced to describe the constraints and objectives. In addition, in order to provide the population more diversity and improve the convergence speed, a learning strategy is embedded in the algorithm framework. Simulation results indicated that by applying the initial guess generated using the VLDE approach, the value of processing time, and Newton iteration times for the NLP solver can be improved significantly. Also, according to the convergency test, the proposed method can have a better performance in terms of convergence ability and convergence time than other methods considered in this paper. Therefore, since the simulation results have proved the feasibility, as well as the superiority, of the proposed algorithm, it is effective and efficient to use the VLDE-based method in trajectory optimization problem. Our future work will focus on improving the stability of the VLDE-based algorithm so that it can be applied in dealing with more complex mission scenarios [38]. Moreover, it would be also worthwhile to enhance the real-time performance of the proposed method so as to employ it for solving real-time trajectory planning problems.

ACKNOWLEDGMENT

The authors would like to thank the industrial sponsor for their help and support of this paper.

REFERENCES

- [1] A. L. Rodriguez-Vazquez, M. A. Martin-Prats, and F. Bernelli-Zazzera, "Spacecraft magnetic attitude control using approximating sequence Riccati equations," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 4, pp. 3374–3385, Oct. 2015.
- [2] J. Zhang, K. Ma, G. Meng, and S. Tian, "Spacecraft maneuvers via singularity-avoidance of control moment gyros based on dual-mode model predictive control," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 4, pp. 2546–2559, Oct. 2015.
- [3] A. Rahimi, K. Dev Kumar, and H. Alighanbari, "Particle swarm optimization applied to spacecraft reentry trajectory," *J. Guid., Control, Dyn.*, vol. 36, no. 1, pp. 307–310, 2012.

- [4] W. Robert, A. Mark, B. Jeffrey, W. Robert, A. Mark, and B. Jeffrey
Minimum heating reentry trajectories for advanced hypersonic launch vehicles
In *Proc. Guid., Navig., Control Conf.*, pp. 354–363, 1997.
- [5] T. R. Jorris and R. G. Cobb
Multiple method 2-D trajectory optimization satisfying waypoints and no-fly zone constraints
J. Guid., Control, Dyn., vol. 31, no. 3, pp. 543–553, 2008.
- [6] T. R. Jorris and R. G. Cobb
Three-dimensional trajectory optimization satisfying waypoint and no-fly zone constraints
J. Guid., Control Dyn., vol. 32, no. 2, pp. 551–572, 2009.
- [7] J. T. Betts
Survey of numerical methods for trajectory optimization
J. Guid., Control, Dyn., vol. 21, no. 2, pp. 193–207, 1998.
- [8] O. A. Yakimenko
Direct method for rapid prototyping of near-optimal aircraft trajectories
J. Guid., Control Dyn., vol. 23, no. 5, pp. 865–875, 2000.
- [9] X. Liu, Z. Shen, and P. Lu
Entry trajectory optimization by second-order cone programming
J. Guid., Control Dyn., vol. 39, no. 2, pp. 227–241, 2016.
- [10] X. Liu and P. Lu
Solving nonconvex optimal control problems by convex optimization
J. Guid., Control, Dyn., vol. 37, no. 3, pp. 750–765, 2014.
- [11] D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao
Direct trajectory optimization and costate estimation via an orthogonal collocation method
J. Guid., Control, Dyn., vol. 29, no. 6, pp. 1435–1440, 2006.
- [12] S. Jain and P. Tsiotras
Trajectory optimization using multiresolution techniques
J. Guid., Control, Dyn., vol. 31, no. 5, pp. 1424–1436, 2008.
- [13] Z. Kenan and C. Wanchun
Reentry vehicle constrained trajectory optimization
In *Proc. 17th AIAA Int. Space Planes Hypersonic Syst. Technol. Conf.*, 2011, pp. 1–16.
- [14] T. L. Hui, Y. J. Ping, F. Qun, and L. J. Jun
Reentry skipping trajectory optimization using direct parameter optimization method
In *Proc. 14th AIAA/AHI Space Planes Hypersonic Syst. Technol. Conf.*, 2006, pp. 824–837.
- [15] J.-P. Y. Lianghui Tu
Reentry trajectory optimization using direct collocation method and nonlinear programming
In *Proc. 57th Int. Astronaut. Congr.*, 2006, pp. 301–309.
- [16] G. Peter and W. Klaus
Trajectory optimization using a combination of direct multiple shooting and collocation
In *Proc. AIAA Guid., Navig., Control Conf. Navig.*, 2001.
- [17] G. W. Reddien
Collocation at Gauss points as a discretization in optimal control
SIAM J. Control Optim., vol. 17, no. 2, pp. 298–306, 1979.
- [18] Z. Duan and X. R. Li
Analysis, design, and estimation of linear equality-constrained dynamic systems
IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 4, pp. 2732–2746, Oct. 2015.
- [19] W. W. Hager
Runge-Kutta methods in optimal control and the transformed adjoint system
Numerische Mathematik, vol. 87, no. 2, pp. 247–282, 2000.
- [20] D. Christopher, H. William, and R. Anil
An improved adaptive hp algorithm using pseudospectral methods for optimal control
In *Proc. AIAA/AAS Astrodyn. Spec. Conf.*, 2010, pp. 1–14.
- [21] C. L. Darby, W. W. Hager, and A. V. Rao
Direct trajectory optimization using a variable low-order adaptive pseudospectral method
J. Spacecraft Rockets, vol. 48, no. 3, pp. 433–445, 2011.
- [22] H. Hongyan, H. William, and R. Anil
Convergence of a Gauss pseudospectral method for optimal control
In *Proc. AIAA Guid., Navig., Control Conf.*, 2012, pp. 1–9.
- [23] W. W. Hager, H. Hou, and A. V. Rao
Convergence rate for a Gauss collocation method applied to unconstrained optimal control
J. Optim. Theory Appl., vol. 169, pp. 1–24, 2016.
- [24] G. Divya, H. William, and R. Anil
Gauss pseudospectral method for solving infinite-horizon optimal control problems
In *Proc. AIAA Guid., Navig., Control Conf.*, 2010, pp. 1–9.
- [25] R. Jeremy
Launch vehicle trajectory optimization using a Legendre pseudospectral method
In *Proc. AIAA Guid., Navig., Control Conf. Exhib.*, 2003, pp. 1–11.
- [26] D. W. Bruno
Generation of pseudospectral differentiation matrices I
SIAM J. Numer. Anal., vol. 34, no. 4, pp. 1640–1657, 1997.
- [27] J. Timothy, S. Christopher, F. Franklin, and R. Anil
Constrained trajectory optimization using pseudospectral methods
In *Proc. AIAA Atmos., Flight, Mech. Conf. Exhib.*, 2008, pp. 1–33.
- [28] S. M. Elsayed, R. A. Sarker, and D. L. Essam
An improved self-adaptive differential evolution algorithm for optimization problems
IEEE Trans. Ind. Informat., vol. 9, no. 1, pp. 89–99, Feb. 2013.
- [29] N. Padhye, P. Mittal, and K. Deb
Feasibility preserving constraint-handling strategies for real parameter evolutionary optimization
Comput. Optim. Appl., vol. 62, no. 3, pp. 851–890, 2015.
- [30] C.-F. Hu and Y. Xin
Reentry trajectory optimization for hypersonic vehicles using fuzzy satisfactory goal programming method
Int. J. Autom. Comput., vol. 12, no. 2, pp. 171–181, 2015.
- [31] A. Rajesh
Reentry trajectory optimization: Evolutionary approach
Proc. 9th AIAA/ISSMO Symp. Multidisciplinary Anal. Optim. Conf., 2002, pp. 1–5.
- [32] Z. Dingni and L. Yi
RLV reentry trajectory optimization through hybridization of an improved GA and a SQP algorithm
In *Proc. AIAA Guid., Navig., Control Conf.*, 2011, pp. 1–10.
- [33] C. Gan, W. Zi-ming, X. Min, and C. Si-Lu
Genetic algorithm optimization of RLV reentry trajectory
In *Proc. Int. Space Planes Hypersonic Syst. Technol. Conf.*, 2005, pp. 1–6.
- [34] D. Garg *et al.*
Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a radau pseudospectral method
Comput. Optim. Appl., vol. 49, no. 2, pp. 335–358, 2011.
- [35] J. A. Nelder and R. Mead
A simplex method for function minimization
Comput. J., vol. 7, no. 4, pp. 308–313, 1965.
- [36] H. Duan and S. Li
Artificial bee colony based direct collocation for reentry trajectory optimization of hypersonic vehicle
IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 1, pp. 615–626, Jan. 2015.
- [37] C. W. Brunner and P. Lu
Skip entry trajectory planning and guidance
J. Guid., Control, Dyn., vol. 31, no. 5, pp. 1210–1219, 2008.
- [38] R. Chai, A. Savvaris, and A. Tsourdos
Fuzzy physical programming for space manoeuvre vehicles trajectory optimization based on hp-adaptive pseudospectral



Runqi Chai (S'15) was born in Beijing, China, in 1993. He received the B.S. degree in information and computing science from the North China University of Technology, Beijing, China, in 2015. He is currently working toward the Ph.D. degree in aerospace engineering at Cranfield University, Cranfield, U.K.

His research interests include trajectory optimization, guidance and control.



Al Savvaris (M'08) received the M.Eng. degree in aerospace systems engineering from the University of Hertfordshire, in 1998, and a Ph.D. degree in radiowave propagation and system design from University of South Wales, in 2004.

He is a Reader in the Centre for Cyber-Physical Systems, Cranfield University, Cranfield, U.K. He established the Autonomous Vehicle Dynamics and Control M.Sc. course and the COMAC training program at Cranfield. He is currently working on InnovateUK funded AirStart and USMOOTH projects. In the past, he worked on the FLAVIIR and ASTRAEA UAS projects, developing new technologies for unmanned systems; working on hardware and system integration. He participated in FP6, in the scope of the FLYSAFE project, working on next generation integrated safety systems. He was a member of the Autonomous Systems National Technical Committee, EPSRC College review member, and a reviewer of several international publications including IMechE and IEEE. He has published more than 50 peer-reviewed journal and conference papers. His research interests and activities include systems integration, hybrid energy management, communication systems, embedded systems, guidance and control.



Antonios Tsourdos (M'99) received the M.Eng. degree in electronic, control and systems engineering from the University of Sheffield, in 1995, M.Sc. degree in systems engineering from Cardiff University, in 1996, and a Ph.D. degree in nonlinear robust missile autopilot design and analysis from Cranfield University, in 1999.

He is currently a Professor of control systems at Cranfield University, Cranfield, U.K. He was appointed as the Head of the Autonomous Systems Group in 2007. He was a member of the Team Stellar, the winning team for the UK MoD Grand Challenge (2008) and the IET Innovation Award (Category Team, 2009). He is an editorial board member on several international publications including IMechE and IEEE. He is a member of the IFAC Technical Committee on Intelligent Autonomous Vehicles, the IET Executive Team on Robotics and Automation, and the ATI Autonomous Systems National Technical Committee. He was also involved in the SEAS DTC on Autonomous Systems Verifications. He has also been engaged in research on guidance and control for single and multiple vehicles as well as verifiable autonomy of autonomous systems and lately dealing with the newly important subjects of integrated system health management and cyber-physical systems. He has published more than 100 peer-reviewed journal and conference papers.

2017-03-09

Violation learning differential evolution-based hp-Adaptive pseudospectral method for trajectory optimization of Space Maneuver Vehicle

Chai, Runqi

Institute of Electrical and Electronics Engineers

Chai R, Savvaris A, Tsourdos A. (2017) Violation learning differential evolution-based hp-Adaptive pseudospectral method for trajectory optimization of space maneuver vehicle. IEEE Transactions on Aerospace and Electronic Systems, Volume 53, Issue 4, August 2017, pp. 2031 - 2044

<http://dx.doi.org/10.1109/TAES.2017.2680698>

Downloaded from Cranfield Library Services E-Repository